浮動小数係数多変数多項式の終結式計算における桁落ち誤差とその解析 (数式処理における理論と応用の研究)

AUTHOR(S):

佐藤, 智之; 佐々木, 建昭

CITATION:

佐藤, 智之 ...[et al]. 浮動小数係数多変数多項式の終結式計算における桁落ち誤差とその解析 (数式処理における理論と応用の研究). 数理解析研究所講究録 1999, 1085: 120-131

ISSUE DATE:

1999-03

URL:

http://hdl.handle.net/2433/62800

数理解析研究所講究録
1085巻 1999年 120-131

浮動小数係数多変数多項式の終結式計算における桁落ち誤差とその解析

筑波大学大学院 教育研究科 佐藤 智之 (Tomoyuki SATO) *
筑波大学 数学系 佐々木 建昭 (Tateaki SASAKI)†

1 はじめに

本稿では、浮動小数係数多変数多項式に対して、それらの終結式を5つの代表的な方法で計算し、その誤差について考察する。浮動小数係数多変数多項式の終結式を計算する場合、用いる算法によって大きな桁落ちが起きたり起きなかったりする。昨年度は4つの方法に対して実験し、いくつかの方法では非常に大きな桁落ちが生じることを実証したが、その理由は明らかでなかった。そこで今回は、昨年発表したの各算法や実験結果を踏まえた上で、さらなる実験を行って桁落ち誤差の原因を解明する。

多変数多項式の終結式計算の手法には、Sylvester行列式を計算、Bezout行列式を計算[KA69]、部分終結式算法で多項式剰余列を計算[BT71]、多項式補間に基ずく、Collinsのアルゴリズム[Col71]、頭項消去法で主係数を消去する[Sas91b]、の5つの方法があげられる。まず第2章では、昨年触れなかったCollinsのアルゴリズムについて触れ、第3章ではそれぞれの算法に対しての実験結果を示す。第4章では部分終結式算法、Collinsのアルゴリズム、Bareissの算法が何故大きな誤差を引き起こすのか、その理由を明らかにする。

2 終結式算法のサーベイ

本章では昨年度触れなかったCollinsのアルゴリズムについて説明する。なお、今回は部分終結式算法[BT71]、Bareissのアルゴリズム[Bar68]、効率的ガウス消去法[SM82]の説明を割愛する。これらの算法については、昨年度の数理研講究録あるいは参考文献を参照されたい。

*tsato@math.tsukuba.ac.jp
†sasaki@math.tsukuba.ac.jp

2.1 Collinsのアルゴリズム

Collinsの算法とは、与式を1変数多項式に写像し、1変数多項式の終結式から答えを補間するという方法である。すなわち、変数 $y,\cdots,z$ に数値 $y_i,\cdots,z_i$ $(i=1,\cdots,N)$ を代入して、N 組の1変数多項式の終結式

$$\mathrm{res}(F(x,y_i,\cdots,z_i),G(x,y_i,\cdots,z_i)),\quad i=1,2,\cdots,N \tag{1}$$

を計算し、これらの終結式(数値)から $y,\cdots,z$ の多項式である終結式を補間する方法である。ここで、補間法としてはLagrange補間法、あるいはNewton補間法がある(詳しくは[Lip71])。N の値は次のように定める。まず、Sylvester行列式かBezout行列式から、求めるべき多項式の $y,\cdots,z$ に関するそれぞれの上限次数 $N_y,\cdots,N_z$ を決める。つまり、次式を満たすような $N_y,\cdots,N_z$ を決める。

$$\deg_y(\mathrm{resultant})\le N_y,\cdots,\deg_z(\mathrm{resultant})\le N_z \tag{2}$$

これら $N_y,\cdots,N_z$ から $N=(N_y+1)\times\cdots\times(N_z+1)$ と定めればよい。

3 実験

我々は、上記の5つのアルゴリズムに対して、これらの算法を数式処理システム**GAL**に実装し、2〜4変数の多項式に対して実験し、行列式の係数に含まれる誤差を調べた。具体的には、有理係数多項式 F_r,G_r とそれらを浮動小数係数に変換した多項式 F_f,G_f に対して、それらの終結式 $\mathrm{res}(F_r,G_r)$ と $\mathrm{res}(F_f,G_f)$ を計算し、計算結果の対応する係数どうしを比較して、$\mathrm{res}(F_f,G_f)$ の各係数に含まれる誤差を測定した。以下において、最大(最小)相対誤差とは、$\mathrm{res}(F_f,G_f)$ の各係数の相対誤差のうち最大(最小)のものを表わす。また、アルゴリズムを次の記号で表すものとする。

resSo	:	部分終結式算法 (Original)
resSi	:	部分終結式算法 (Improved)
resCr	:	Collinsのアルゴリズム (補間点は実軸上)
resCc	:	Collinsのアルゴリズム (補間点は複素平面上)
resM	:	小行列展開法
resG	:	ガウス消去法
resEG	:	効率的ガウス消去法

3.1 Collinsのアルゴリズム

Collinsのアルゴリズムにおいての桁落ち誤差は補間点に大きく依存する。そこで補間点を実軸上に選ぶ場合と、複素平面上に選ぶ場合を比較した。

例 **2-1**　$R_{2,1}$=res$(F_{2,1}/3.0, G_{2,1}/3.0)$

$$\begin{cases} F_{2,1} = (y+1)x^4 + (2y^2-3)x^2 + (y^2+y-3)x - (y^3-2y^2+y) \\ G_{2,1} = (y^2+2y-1)x^3 + (y-1)^2x + (y^3-3y^2+y+5) \end{cases}$$

$$\deg(R_{2,1}) = 17, \quad 0.00045 < |\text{coeff}(R_{2,1})| < 6.17.$$

$R_{2,1}$ の上限次数は Bezout の行列式で計算すると 20 なので、補間点を $(y_0, y_1, y_2, \cdots, y_{20})$ とし、補間数を実数、複素数の場合に分けて、次のように定めた。ただし、r, c は実数とする。

$$\begin{cases} (-10r, -9r, \cdots, -r, 0, +r, \cdots, +9r, +10r) \\ (-5c, -(4 \pm i)c, \cdots, -(1 \pm 4i)c, (0 \pm 5i)c, +(1 \pm 4i)c, \cdots, +(4 \pm i)c, +5c) \end{cases} \tag{3}$$

r	min-rel-error	max-rel-error	c	min-rel-error	max-rel-error
1.0	1.0×10^{-16}	3.6×10^{-07}	1.0	3.6×10^{-16}	7.1×10^{-10}
0.5	1.0×10^{-16}	1.1×10^{-11}	0.5	1.0×10^{-16}	8.8×10^{-14}
0.3	2.9×10^{-15}	2.0×10^{-06}	0.3	5.5×10^{-16}	2.9×10^{-12}
0.2	3.8×10^{-13}	1.3×10^{-09}	0.2	3.5×10^{-16}	8.8×10^{-11}
0.1	2.8×10^{-13}	3.5×10^{-06}	0.1	4.2×10^{-16}	4.7×10^{-06}

表 1: 補間点の違いによる最大最小相対誤差

表 1 は Lagrange の補間公式を用い、$r, c = 1.0, 0.5, 0.4, 0.3, 0.2, 0.1$ と補間点を定め、Collins のアルゴリズムで計算したときの係数の最大最小相対誤差である。それぞれのパラメータに対して、各係数は次数が低い項ほど正確である。また、Newton の補間公式も用いてみたが、誤差は表 1 の値よりもはるかに大きくなった。

3.2　2 変数多項式の場合

ここでは、2 変数多項式に対する実験結果を与える。なお、Collins のアルゴリズムでは Lagrange の補間公式を使い、上記の実験結果を基に $r = c = 0.5$ とした。

例 **2-2**　$R_{2,2}$ =res$(F_{2,2}/7.0, G_{2,2}/7.0)$

$$\begin{cases} F_{2,2} = (y+2)^3x^5 - (y-1)^3x^4 + (y+2)^3x^3 + (y-2)^3x^2 - (y^2+4y-5)x \\ \qquad\qquad -(y^3+2y^2+5) \\ G_{2,2} = (2y^2-1)^2x^4 + (y^3+4y^2-3)x^3 + (2y^2-3)x^2 + (y^2+y-3)^2x \\ \qquad\qquad -(y^3-2y^2+y) \end{cases}$$

$$\deg(R_{2,2}) = 32, \quad 0.00019 < |\text{coeff}(R_{2,2})| < 339.$$

例 **2-3**　$R_{2,3}$ =res$(F_{2,3}/3.0, G_{2,3}/3.0)$

$$\begin{cases} F_{2,3} = (2y^2-1)^2x^6 + (2y^3-3)^2x^5 - (y^2+1)^2x^4 + (y^3+y^2-1)x^2 \\ \qquad\qquad -(2y^3-y+4)x + (y^3-3y^2+y+5) \\ G_{2,3} = (2y+1)^5x^6 - (y-2)^5x^5 + (2y^2+y-4)^2x^4 - (y^2+2y-1)^2x^2(y^2-5)^2x \\ \qquad\qquad +(y^4+3y^3-2y+7) \end{cases}$$

$$\deg(R_{2,3})= 50, \quad 0.0016 < |\text{coeff}(R_{2,3})| < 1.4 \times 10^{12}.$$

method	例 2-1	time	例 2-2	time	例 2-3	time
resSo	1.8×10^{-04}	60	$\cdots$(1)	132	$\cdots$(5)	2000
resSi	1.7×10^{-11}	35	$\cdots$(2)	266	$\cdots$(6)	1270
resCr	1.1×10^{-11}	100	$\cdots$(3)	434	$\cdots$(7)	1860
resCc	8.8×10^{-14}	250	2.1×10^{-07}	1253	$\cdots$(8)	5120
resM	1.1×10^{-15}	20	2.6×10^{-15}	309	3.2×10^{-15}	2010
resG	4.4×10^{-11}	33	$\cdots$(4)	223	$\cdots$(9)	1060
resEG	5.0×10^{-16}	26	3.8×10^{-15}	520	2.3×10^{-14}	2740

表 2: それぞれの算法に対する最大相対誤差と計測時間 (ミリ秒)
ただし、"$\cdots$"は、どれかの係数が誤差だらけとなる場合で、
誤差だらけの係数の個数は以下のとおりである。
(1)23, (2)11, (3)6, (4)15, (5)39, (6)36, (7)44, (8)30, (9)23

表 2 は 5 つの算法 (細かく分類すると 7 つの算法) で生じた、最大相対誤差及びタイミングデータを表わしている。この中で、改良していない部分終結式算法が大きな桁落ちを起こしていることが分かる。改良した部分終結式算法や、Collins のアルゴリズム、ガウスの消去法は、改良していない部分終結式算法ほど桁落ちを起こしていない。しかしながら、問題のサイズが大きくなってくると、これらも大きな桁落ちを起こす。ここでも、小行列展開法、効率的ガウス消去法はほとんど桁落ちを起こさない。

3.3　3〜4 変数多項式の場合

ここでは、小行列展開法、ガウスの消去法、効率的ガウス消去法の各算法に対して実験をした。

例 **3-1**　$R_{3,1} = \text{res}(F_{3,1}/3.0, G_{3,1}/3.0)$

$$\begin{cases} F_{3,1} = (y-z+1)^2x^4 + (2y^2-3z^2)x^3 - (y^2-3yz)x^2 + (y^2+y-3z^2)x \\ \qquad\qquad -(y^3-2y^2z+yz^2) \\ G_{3,1} = (y^2+2yz-1)x^3 + (y+z-1)^2x^2 - (yz+z^2+z)x \\ \qquad\qquad +(y^3+3y^2-yz^2+5z^3) \end{cases}$$

$$\#\text{terms}(R_{3,1}) = 173.$$

例 3-2 $R_{3,2} = \text{res}(F_{3,2}/7.0, G_{3,2}/7.0)$

$$\begin{cases} F_{3,2} = (y-z+3)^2x^6 + (2y^2-3z^2)x^5 - (y^2+3yz)x^4 + (y+2x+3)^2x^2 \\ \qquad\qquad +(yz-3y+2z)x - (y^2+z^2-5) \\ G_{3,2} = (y^2+2yz-3)x^4 + (y+z-5)^2x^3 - (y^2+2yz)x^2 - (yz+z^2+5z)x \\ \qquad\qquad +(2y^2-3yz^2+5z^3+8) \end{cases}$$

$$\#\text{terms}(R_{3,2}) = 304.$$

例 4-1 $R_{4,1} = \text{res}(F_{4,1}/3.0, G_{4,1}/3.0)$

$$\begin{cases} F_{4,1} = (y+z-u+1)^2x^4 + (2y^2+z^2-2u^2)x^3 - (y^2-2yz+3zu)x^2 \\ \qquad\qquad +(y^2+3z^2-2zu-2u^2)x - (y^3-2y^2z+yz^2+2z^2u-3z^3+u^3) \\ G_{4,1} = (y^2+2yz+2zu-1)x^3 + (y+z-u-1)^2x^2 - (yz+z^2-2u^2+z-u+1)x \\ \qquad\qquad +(y^3+3y^2-yz^2+2z^3-3zu^2+u^2-z+u) \end{cases}$$

$$\#\text{terms}(R_{4,1}) = 1275.$$

例 4-2 $R_{4,2} = \text{res}(F_{4,2}/7.0, G_{4,2}/7.0)$

$$\begin{cases} F_{4,2} = (y-z+u+1)^2x^5 + (y^2-3z^2+2u^2)x^4 - (y^2-yz+2yu-2zu)x^3 \\ \qquad\qquad +(yz+z^2-2u^2)x^2 - (3yz+2z^2-zu^2)x - (y^3-2y^2z+yz^2+2z^2u) \\ G_{4,2} = (y^2+2yz+yu-z^2-3)x^4 + (y-z+2u-2)^2x^3 \\ \qquad\qquad -(y^2+2yz-z^2-2zu+2z)x^2 + (yz-3y+2z-u^2)x \\ \qquad\qquad -(y^2-2yz+z^2-zu+2u^2-2z+2u-5) \end{cases}$$

$$\#\text{terms}(R_{4,2}) = 2201.$$

method	例 3-1 time	例 3-2 time	例 4-1 time	例 4-2 time
resM	4.4×10^{-15} 0.32	4.3×10^{-14} 6.40	9.1×10^{-14} 4.01	1.3×10^{-13} 21.60
resG	7.3×10^{-07} 0.81	$\cdots$ 6.41	8.8×10^{-04} 13.78	$\cdots$ 243.00
resEG	4.4×10^{-15} 0.32	5.7×10^{-14} 12.70	1.8×10^{-13} 9.15	5.9×10^{-13} 70.00

表 3: 3 つの方法における最大相対誤差と計測時間 (秒)
ただし、"..."は誤差だらけの係数が現れたことを意味する。

表 3 は小行列展開法、ガウス消去法、効率的ガウス消去法の最大相対誤差及びタイミングデータを表わしている。驚くべき事は、小行列展開法がきわめて効率が良いことである。しかしながら、今回試した方法では、Bezout の行列式のオーダーが小さい。もしオーダーをあげると効率的ガウス消去法のデータの方が良くなるかもしれない。しかし、現実的な計算機の性能や時間等を考えると、小行列展開法が最良といえる。

3.4 大きな桁落ちが不可避的に生じる場合

以上の結果から、小行列展開法や効率的ガウス消去法は大きな誤差を生まないと考えがちではあるが、入力多項式が近似的に共通の因子を持っている場合、大きな誤差を生んでしまう。これは避けられないことである。このことを具体的な例で示す。

例 5-1 $R_{2,4}$=res$(F_{2,4}/3.0, G_{2,4}/3.0)$

$$\begin{cases} F_{2,4} = (x+y-1) \times G_{2,1} + F_{2,1}/1000 \\ G_{2,4} = (x-2y+1) \times G_{2,1} \end{cases}$$

$$\deg(R_{2,4})=23, \quad 3.6 \times 10^{-15} < |\mathrm{coeff}(R_{2,4})| < 2.6 \times 10^{-7}.$$

例 5-2 $R_{2,5}$=res$(F_{2,5}/7.0, G_{2,5}/7.0)$

$$\begin{cases} F_{2,5} = (x+y-1) \times G_{2,2} + F_{2,2}/1000 \\ G_{2,5} = (x-2y+3) \times G_{2,2} \end{cases}$$

$$\deg(R_{2,4})=41, \quad 5.4 \times 10^{-15} < |\mathrm{coeff}(R_{2,5})| < 3.8 \times 10^{-7}.$$

method 例	resM min-rel-error	resM max-rel-error	resEG min-rel-error	resEG max-rel-error
例 5-1	4.4×10^{-13}	5.7×10^{-8}	6.1×10^{-13}	4.0×10^{-8}
例 5-2	2.0×10^{-12}	4.4×10^{-9}	3.3×10^{-10}	7.7×10^{-3}

表 4: 入力多項式が近似的に共通の因数を持っている場合。

表 4 は小行列展開法、効率的ガウス消去法で引き起こされる最大・最小相対誤差を表わしている。この例では効率的ガウス消去法よりは小行列展開法の結果の方が良い。

4 大きな桁落ち誤差の原因

ここでは、部分終結式算法、Collins のアルゴリズム、部分終結式 PRS アルゴリズムが何故大きな桁落ちを起こすのか、解明する。まず、次のケースで具体的に説明する。$\deg_x(F) =$

$n+1, \deg_x(G) = n$:

$$F = f_{n+1}x^{n+1} + f_n x^n + \cdots + f_0, \quad G = g_n x^n + g_{n-1}x^{n-1} + \cdots + g_0. \tag{4}$$

このとき、擬剰余 $P_3 \equiv \text{remainder}(g_n^2 F, G)$ は次の式で求められる。

$$\begin{aligned} P_3 &= g_n^2 F - (g_n f_{n+1} x + g_n f_n - g_{n-1} f_{n+1}) G \\ &= \begin{vmatrix} g_n & g_{n-1} & g_{n-2} \\ & g_n & g_{n-1} \\ f_{n+1} & f_n & f_{n-1} \end{vmatrix} x^{n-1} + \cdots + \begin{vmatrix} g_n & g_{n-1} & g_{i-1} \\ & g_n & g_i \\ f_{n+1} & f_n & f_i \end{vmatrix} x^i + \cdots . \end{aligned} \tag{5}$$

次に、$\tilde{P}_4 \equiv \text{remainder}(\text{lc}(P_3)^2 G, P_3)$ は次の式で表わされる。

$$\tilde{P}_4 = \sum_{i=0}^{n-2} \begin{vmatrix} \begin{vmatrix} g_n & g_{n-1} & g_{n-2} \\ & g_n & g_{n-1} \\ f_{n+1} & f_n & f_{n-1} \end{vmatrix} & \begin{vmatrix} g_n & g_{n-1} & g_{n-3} \\ & g_n & g_{n-2} \\ f_{n+1} & f_n & f_{n-2} \end{vmatrix} & \begin{vmatrix} g_n & g_{n-1} & g_{i-2} \\ & g_n & g_{i-1} \\ f_{n+1} & f_n & f_{i-1} \end{vmatrix} \\ \begin{vmatrix} g_n & g_{n-1} & g_{n-2} \\ & g_n & g_{n-1} \\ f_{n+1} & f_n & f_{n-1} \end{vmatrix} & \begin{vmatrix} g_n & g_{n-1} & g_{i-1} \\ & g_n & g_i \\ f_{n+1} & f_n & f_i \end{vmatrix} & \\ g_n & g_{n-1} & g_i \end{vmatrix} x^i. \tag{6}$$

x^i 項の行列の係数を計算すると次のようになる。

$$\begin{aligned} & g_n^5 \times (-f_{n-1}f_{i-1} + f_{n-2}f_i) \\ + \; & g_n^4 \times (g_{n-1}f_n f_{i-1} - g_{n-1}f_{n-1}f_i + g_{n-2}f_{n+1}f_{i-1} - g_{n-2}f_n f_i - g_{n-3}f_{n+1}f_i \\ & \quad - g_i f_n f_{n-2} + g_i f_{n-1}^2 - g_{i-1}f_{n+1}f_{n-2} + g_{i-1}f_n f_{n-1} + g_{i-2}f_{n+1}f_{n-1}) \\ + \; & g_n^3 \times (-g_{n-1}^2 f_{n+1}f_{i-1} + g_{n-1}^2 f_n f_i + 2g_{n-1}g_{n-2}f_{n+1}f_i + g_{n-1}g_i f_{n+1}f_{n-2} \\ & \quad - g_{n-1}g_i f_n f_{n-1} - g_{n-1}g_{i-1}f_n^2 - g_{n-1}g_{i-2}f_{n+1}f_n - 2g_{n-2}g_i f_{n+1}f_{n-1} \\ & \quad + g_{n-2}g_i f_n^2 - g_{n-2}g_{i-2}f_{n+1}^2 + g_{n-3}g_i f_{n+1}f_n + g_{n-3}g_{i-1}f_{n+1}^2) \\ + \; & g_n^2 \times (-g_{n-1}^3 f_{n+1}f_i + g_{n-1}^2 g_i f_{n+1}f_{n-1} + g_{n-1}^2 g_{i-1}f_{n+1}f_n + g_{n-1}^2 g_{i-2}f_{n+1}^2 \\ & \quad - g_{n-1}g_{n-2}g_i f_{n+1}f_n - g_{n-1}g_{n-2}g_{i-1}f_{n+1}^2 - g_{n-1}g_{n-3}g_i f_{n+1}^2 + g_{n-2}^2 g_i f_{n+1}^2) \\ + \; & g_n^1 \times (-\underline{g_{n-1}^3 g_{i-1}f_{n+1}^2} + \underline{g_{n-1}^2 g_{n-2}g_i f_{n+1}^2} + \underline{2g_{n-1}^3 g_i f_{n+1}f_n} \\ & \quad + \underline{g_{n-1}^3 g_{i-1}f_{n+1}^2} + \underline{g_{n-1}^2 g_{n-2}g_i f_{n+1}^2} - \underline{2g_{n-1}^3 g_i f_{n+1}f_n} - \underline{2g_{n-1}^2 g_{n-2}g_i f_{n+1}^2} \\ + \; & g_n^0 \times (-\underline{g_{n-1}^4 g_i f_{n+1}^2} + \underline{g_{n-1}^4 g_i f_{n+1}^2}). \end{aligned} \tag{7}$$

下線部の項 ($g_n^1 \times (\cdots)$ と $g_n^0 \times (\cdots)$) は、互いにキャンセルして計算結果には現われないものであるが、計算途中では現われる。

多項式剰余列を部分終結式算法で計算すると上記下線部の項のように互いにキャンセルする項が現われる。したがって (7) 式の下線部の項が他の項よりも大きい場合、大きな桁落ち誤差が起きると考えられる。しかしながら、例題 2-1 の場合にはそうではない: 消去された項を調べたところ、その係数部は消去されなかった項と同じ程度の大きさであった。したがって、上記の項のキャンセルが大きな桁落ちの原因ではない。

4.1 多項式除算における誤差拡大

部分終結式算法において大きな桁落ち誤差が起きる理由は、多項式除算における誤差の拡大である。特に、他の項に比べ微小な係数をもつ多項式で割った場合の大きな誤差を生ずる。そこで、次のような y に関する多項式 C と D を考える。

$$C = c_l y^l + \cdots + c_0, \quad D = d_m y^m + \cdots + d_0, \quad (l > m). \tag{8}$$

D による C の除算は $y^i (i = l, l-1, \cdots, m)$ 項の消去である。y^l 項の消去は

$$C' := C - (c_l/d_m) y^{l-m} \times D \tag{9}$$

である。ただし、$\mathrm{quotient}(C, D) = (c_l/d_m) y^{l-m} + \cdots$ となる。いま、$C = QD + R$, $\|C\| = \|D\| = 1$, $\|Q\| \simeq 1$, $0 < |d_m| \ll 1$ とする (実際、上記の実験例ではそうなっている)。このとき、多項式除算途中における被除多項式の主係数を $C'_{l'} = c'_{l'} y^{l'} + \cdots + c'_0$ とすると、商 Q の $y^{l'-m}$ の係数は $c'_{l'}/d_m$ となる。ここで、$\|Q\| \simeq 1$ なので、$|c'_{l'}/d_m| \lesssim 1$ がすべての l' に対して成立する。よって、$|c'_{l'}| \lesssim |d_m| \ll 1$ となるが、C に元々に含まれていた誤差の方は小さくなることはない。したがって C の $y^{l'}$ 項の大きさが 1 程度だった場合、$C'_{l'}$ の主係数の相対誤差は $|1/d_m|$ 倍程度になっているはずである。

表 5 は $\mathrm{res}(F_{2,1}/3.0, G_{2,1}/3.0)$ を計算したときの剰余の最大相対誤差である。

remainders	original	improved
$P_1 = F_{2,1}/3.0$		
$P_2 = G_{2,1}/3.0$		
$P_3 = \mathrm{prem}(P_1, P_2)$	1.0×10^{-16}	1.0×10^{-16}
$\tilde{P}_4 = \mathrm{prem}(P_2, P_3)$	4.6×10^{-13}	1.0×10^{-16}
$P_4 = \tilde{P}_4 \div \mathrm{lc}(P_2)^2$	7.0×10^{-12}	1.0×10^{-16}
$\tilde{P}_5 = \mathrm{prem}(P_3, P_4)$	9.1×10^{-04}	4.0×10^{-11}
$P_5 = \tilde{P}_4 \div \mathrm{lc}(P_3)^2$	1.8×10^{-04}	1.7×10^{-11}

表 5: 部分終結式算法での剰余の最大相対誤差
ただし、"prem"は擬剰余を表わす。

ここで $\tilde{P}_5$ について詳しく説明する。いま表 5 において次のように P_3, P_4 をおく。

$$P_3 = C_2(y)x^2 + C_1(y)x + C_0(y), \quad P_4 = D_1(y)x + D_0(y) \tag{10}$$

このとき、P_3, P_4 の係数は実際には次のようになっている。

$$\begin{aligned} C_2 &= 0.076\cdots y^6 + \cdots, & \|C_2\| &\simeq 0.59 \\ D_1 &= 0.0082\cdots y^{11} + \cdots, & \|D_1\| &\simeq 1.9 \\ D_1^2C_2 &= 0.0000050\cdots y^{28} + \cdots, & \|D_1^2C_2\| &\simeq 5.0 \end{aligned}$$

$\mathrm{prem}(P_3, P_4)$ の計算では、次数 28 次の多項式 $D_1^2C_2$ を主係数が 0.0082 である多項式 D_1 で割る除算が行われる。よって上述の説明のように、商と剰余に大きな誤差が含まれてしまうのである。なお、剰余列を計算するにしたがい主係数が相対的に小さくなることは、常に成立するわけではない。しかしながら、多項式剰余列の計算では主係数のべき乗 (一般的には 2 乗) を次々とかけるので、確率的に主係数が小さくなるのである。

4.2 部分終結式算法の工夫 (Improved)

擬剰余計算をちょっと工夫することで、誤差を減らすことができる。擬剰余計算は除数で主項を除くという事にほかならないので、この方法を使わないで消去すれば良いと考えられる。例えば、(10) の式での P_3 と P_4 において、P_3 の x^2 項は P_4 で次のように消去される。

$$P_3' := D_1P_3 - xC_2P_4 = (D_1C_1 - C_2D_0)x + D_1C_0 \tag{11}$$

さらに P_3' の x 項を消去すると、$\mathrm{prem}(P_3, P_4)$ を次のように計算できる。

$$\mathrm{prem}(P_3, P_4) := D_1P_3' - \mathrm{lc}(P_3')P_4 \tag{12}$$

この方法は除算を必要としないため、$\tilde{P}_5$ は大きな桁落ちを生じることなく計算できる。実際に今回の実験では、表 5 に示したように、終結式計算における最大相対誤差が 1.4×10^{-11} であった。しかしながら、$\tilde{P}_4$ と $\tilde{P}_5$ からそれぞれ P_4 と P_5 を求める際の除算にはこの方法は使えない。

4.3 Collins のアルゴリズム

Collins のアルゴリズムにおける Lagrange の補間式を考える。いま、補間点を $(y_0, y_1, \cdots, y_N)$, 標本点を $(v_0, v_1, \cdots, v_N)$ とする。このとき、多項式 $P(y)$ の上限次数を $P(y) \leq N$ とすると、多項式 $P(y)$ は次の式で表わせる。

$$P(y) = \sum_{i=0}^{N} v_i \frac{\prod_{j=0,\neq i}^{N}(y - y_j)}{\prod_{j=0,\neq i}^{N}(y_i - y_j)} \tag{13}$$

ここで、$P(y_i)=v_i$ $(i=0,1,\cdots,N)$ である。$Q(y)$ を次のように定義すると $P(y)$ は (15) 式のように表せる。

$$Q(y)=\prod_{j=0}^{N}(y-y_j),\quad Q'(y)=\frac{dQ(y)}{dy} \tag{14}$$

$$P(y)=\sum_{i=0}^{N}v_iP_i(y),\quad \text{ただし、}P_i(y)=\frac{Q(y)/(y-y_i)}{Q'(y_i)} \tag{15}$$

多項式 $P_i(y)$ は補間点の選び方に大きく左右される。例えば、次の A,B,C のようにそれぞれ補間点を選んだ場合の多項式 P_{10},P_{15},P_{20} は次のようになる。

$$\begin{aligned}
&A:(y_0,y_1,\cdots,y_{20})=(-10.,-9.0,-8.0,\cdots,0,\cdots,9.0,10.)\\
&B:(y_0,y_1,\cdots,y_{20})=(-3.0,-2.7,-2.4,\cdots,0,\cdots,2.7,3.0)\\
&C:(y_0,y_1,\cdots,y_{20})=(-1.0,-0.9,-0.8,\cdots,0,\cdots,0.9,1.0)
\end{aligned}$$

$$A\begin{cases}P_{10}=7.59\cdots10^{-14}y^{20}+\cdots-6.27\cdots10^{-4}y^{10}+\cdots+1\\ P_{15}=-6.37\cdots10^{-15}y^{20}+\cdots+2.56\cdots10^{-5}y^{10}+\cdots+0.0167\cdots y\\ P_{20}=4.11\cdots10^{-19}y^{20}+\cdots-5.62\cdots10^{-10}y^{10}+\cdots-5.41\cdots10^{-7}y\end{cases}$$

$$B\begin{cases}P_{10}=0.00217\cdots y^{20}+\cdots-106.0\cdots y^{10}+\cdots+1\\ P_{15}=-1.82\cdots10^{-4}y^{20}+\cdots+4.34\cdots y^{10}+\cdots+0.0559\cdots y\\ P_{20}=1.17\cdots10^{-8}y^{20}+\cdots-9.51\cdots10^{-5}y^{10}+\cdots-1.80\cdots10^{-6}y\end{cases}$$

$$C\begin{cases}P_{10}=7.59\cdots10^{6}y^{20}+\cdots-6.27\cdots10^{6}y^{10}+\cdots+1\\ P_{15}=-6.37\cdots10^{5}y^{20}+\cdots+2.56\cdots10^{5}y^{10}+\cdots+0.167\cdots y\\ P_{20}=41.1\cdots y^{20}+\cdots-5.62\cdots y^{10}+\cdots-5.41\cdots10^{-6}y\end{cases}$$

Collins の方法とは、このように値が大きくバラつく係数をもつ多項式どうしを加えて、目的の多項式を構成するもので、次数が大きいほど大きな桁落ちを発生してしまう。

4.4 Bareiss のアルゴリズム

最後に、Bezout 行列式に対するガウスの消去法で引き起こされる桁落ち誤差について考える。いま M を $n\times n$ 行列、$M^{(k)}$ を $(n-k)\times(n-k)$ 行列とし、その ij 要素を $M_{ij}^{(k)}$ とする (ただし、i と j は 1 から $n-k$ まで動くのではなく、$k+1$ から n まで動くとする)。ガ

ウスの消去法において、2回目の消去後の要素を $\tilde{M}_{i,j}^{(2)}$ とすると $\tilde{M}_{i,j}^{(2)}$ は次のようになる。

$$
\begin{aligned}
\tilde{M}_{i,j}^{(2)} &= \begin{vmatrix} \begin{vmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \end{vmatrix} & \begin{vmatrix} a_{1,1} & a_{1,j} \\ a_{2,1} & a_{2,j} \end{vmatrix} \\ \begin{vmatrix} a_{1,1} & a_{1,2} \\ a_{i,1} & a_{i,2} \end{vmatrix} & \begin{vmatrix} a_{1,1} & a_{1,j} \\ a_{i,1} & a_{i,j} \end{vmatrix} \end{vmatrix} \\
&= a_{1,1}^2 \times (a_{2,2}a_{i,j} - a_{i,2}a_{2,j}) \\
&- a_{1,1}^1 \times (a_{2,2}a_{i,1}a_{1,j} + a_{1,2}a_{2,1}a_{i,j} - a_{i,2}a_{2,1}a_{1,j} - a_{i,1}a_{1,2}a_{2,j}) \\
&+ a_{1,1}^0 \times (\underline{a_{1,2}a_{2,1}a_{i,1}a_{1,j}} - \underline{a_{i,1}a_{1,2}a_{2,1}a_{1,j}})
\end{aligned}
\tag{16}
$$

$\tilde{M}_{i,j}^{(2)}$ において下線部の項は互いにキャンセルする。このような項のキャンセレーションは、最初の消去を除き、毎回の消去で発生する。いま、$(k+1)$ 回目の消去後の要素を $\tilde{M}_{ij}^{(k+1)}$ とする (ただし、i,j,k は $1 \leq k \leq n-2$ かつ $k+2 \leq i,j \leq n$ を満たすものとする)。$(k+1)$ 回目の消去では、$M_{kk}^{(k-1)}$ を因数としてもつ $\tilde{M}_{ij}^{(k+1)}$ が計算できる。しかしながら、計算途中では各項はバラバラに計算され、$\tilde{M}_{ij}^{(k+1)}$ は因数 $M_{kk}^{(k-1)}$ を含む形で計算されるわけではない。したがって、そのような項は互いにキャンセルされなければならない。このキャンセレーションから誤差が発生する可能性がある。しかしながら、実際にはそのキャンセレーションは大きな誤差を引き起こすわけではない。表2における誤差は、$\tilde{M}_{ij}^{(K+1)}$ を $M_{kk}^{(k-1)}$ で割る多項式除算の際に生じる誤差拡大によるものである。なお、剰余列計算に比べると $M_{kk}^{(k-1)}$ の主係数が小さくなる確率は少ない。そのため、ガウス消去法における桁落ち誤差は、剰余列計算における誤差よりも小さいのである。

5 結論

我々は、浮動小数係数多変数多項式の部分終結式に関して、5つの算法を試して、大きな桁落ちが起きる場合を分析した結果、次のような結論を得た。

1) 部分終結式算法においては、微小係数をもつ多項式で高次の多項式で割った場合に大きな桁落ちが起きる。

2) Collins のアルゴリズムが示すように、高次な多項式を補間すると大きな桁落ちが発生する。

3) 小行列展開法や、効率的ガウス消去法が示すように、入力された多項式が近似因数を持たないとき、多項式除算を必要としないアルゴリズムを用いればほとんど桁落ちが起きない。

4) 求める終結式が極端に大きな多項式でなければ、小行列展開法、効率的ガウス消去法は妥当な方法である。

参考文献

[Bar68] E. H. Bareiss, Sylvester's identity and multistep integer-preserving Gaussian elimination. Math. Comput. **22** (1968), 565-578.

[BT71] W. S. Brown and J. F. Traub, On Euclid's algorithm and the theory of subresultants. J. ACM, **18** (1971), 505-514.

[Col67] J. E. Collins, Subresultant and reduced polynomial remainder sequence. J. ACM, **14** (1967), 128-142.

[Col71] J. E. Collins, The calculation of multivariate polynomial resultants. J. ACM, **18** (1971), 515-532.

[KA69] S. Y. Ku and R. J. Adler, Computing polynomial resultants: Bezout's determinant vs. Collins' reduced P.R.S. algorithm. Comm. ACM, **12** (1969), 23-30.

[Lip71] J. D. Lipson, Chinese remainder and interpolation algorithms. Proc. SYMSAC, ACM, (1971), 372-391.

[Sas81] 佐々木 建昭, 数式処理, 情報処理叢書, 情報処理学会, 1981.

[Sas91a] T. Sasaki, Formula manipulation system GAL, *Proc. on Computing in High Energy Physics '91*, Universal Academic Press, Inc. (Tokyo), (1991) 381-389 1991.

[Sas91b] 現代数理科学事典 (編集代表・広中平祐), XVIII 基礎数理 (I), [4] 計算機代数, (佐々木建昭執筆), 大阪書籍, 1991.

[SM82] T. Sasaki and H. Murao, Efficient Gaussian elimination method and linear systems. ACM Trans. Math. Software **8** (1982), 277-289.